Leaderless Byzantine Fault-Tolerance for Blockchains Consensus

***Abstract*—**

I. Introduction

Blockchain, a distributed ledger shared among disparate parties, is gaining popularity since it makes digital transactions possible without a central authority. This great promise has fueled a number of innovations such as cryptocurrencies [4] and smart contracts [1]. One issue that hinders the further deployment of blockchain-based applications is their low throughput and high latency. For example, Bitcoin [4] can only handle $\sim$7 transactions per second (TPS) and each transaction requires one hour to get confirmed. Ethereum [1] also requires several minutes to confirm a transaction. This is due to the fact that they use proof-of-work (PoW) as their consensus layer, which is slow and consumes large amounts of energy.

The blockchain community has been looking for ways to effectively integrate traditional Byzantine fault-tolerant (BFT) protocols into the blockchain consensus layer. For example, IBM's Hyperledger/Fabric blockchain [] currently relies on PBFT [2] for consensus. Leader-based BFT (e.g., PBFT and its successors [**? ? ? ? ?**]) achieves better response latency and throughput in normal case when the leader functions correctly. However, they can only scale to few tens of nodes, due to the extensive coordination especially when the leader is faulty. Furthermore, such protocols rely on the *weak synchrony* assumption (i.e., messages are guaranteed to be delivered after a time-varying bound) to make progress (i.e., *liveness*). It has been criticized that their liveness properties can fail completely when the expected timing assumptions are violated [3]. Therefore, some blockchain researchers have switched to explore asynchronous BFT protocols [**? ? ? ?**] which can make progress as long as the messages are guaranteed to be delivered *eventually* but no other timing assumption is made. However, such asynchronous BFT protocols are typically considered expensive. Even through some efforts have been put on improving such protocols, their performance is still far from practical. For example, the best asynchronous BFT protocol can only scale to 91 nodes, with a latency of 1.5 minutes and a throughput of less than 50 TPS [].

In this paper, we observe that most of the weakly synchronous protocols are leader-based; and the attack that compromises the liveness properties of such protocols is in fact targeting the leader. To this end, we aim to prevent this attack without sacrificing performance by proposing a leaderless BFT in weakly synchronous settings. For many decades, researchers have been focusing on designing either leaderless BFT in asynchronous settings or leader-based BFT in weakly synchronous settings. Leaderless BFT protocols in weakly synchronous setting have been somewhat overlooked. We start from a recently proposed leaderless and synchronous BFT protocol called Snowball [5]. We study its behaviour in both synchronous and weakly synchronous settings. Then, we turn it into a protocol that works in weakly synchronous settings, and add several optimizing techniques including transaction pipeline, block compression and message aggregation.

II. Background and Preliminaries

A. Leader-based BFT

normal case
view-change
attack for PBFT in honeybadger

B. Snowball

C. Message aggregation

Multisignature
binomial swap forest

III. Overview

IV. Conclusion and Future Work

References

[1] Vitalik Buterin. A Next-Generation Smart Contract and Decentralized Application Platform, Accessed in January 2019. https://github.com/ethereum/wiki/wiki/White-Paper.

[2] Miguel Castro and Barbara Liskov. Practical Byzantine Fault Tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, OSDI '99, pages 173–186, Berkeley, CA, USA, 1999. USENIX Association.

[3] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. The Honey Badger of BFT Protocols. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 31–42, New York, NY, USA, 2016. ACM.

[4] Satoshi Nakamoto. Bitcoin: A Peer-to-peer Electronic Cash System, 2011. http://www.bitcoin.org/bitcoin.pdf.

[5] Team Rocket. Snowflake to avalanche: A novel metastable consensus protocol family for cryptocurrencies, 2018. https://avalanchelabs.org/avalanche.pdf.